Predicting adverse outcomes in chronic kidney disease using machine learning methods: data from the modification of diet in renal disease

Zeid Khitan MD[1], Anna P. Shapiro MD[2], Preeya T. Shah MS[1], Juan Sanabria MD[1], Prasanna Santhanam MD[3,1], Komal Sodhi MD[1], Nader G. Abraham PhD[4,1], Joseph I. Shapiro MD[1]

Author Affiliation:

1. Marshall University Joan C. Edwards School of Medicine, Huntington, West Virginia
2. The Cleveland Clinic Foundation
3. John's Hopkins University
4. New York Medical College

Corresponding Author:

Joseph I. Shapiro MD
Joan C. Edwards School of Medicine
Huntington, West Virginia
Email: shapiroj@marshall.edu

Acknowledgments:

The authors wish to acknowledge support from the National Institutes of Health (HL109015, HL071556 and HL105649 to J.I.S, HL34300 to N.G.A) as well as the Brickstreet Foundation and the Huntington Foundation. The authors also wish to acknowledge access to the MDRD dataset furnished by the National Institutes of Health (NIDDK). The MDRD study was conducted by the MDRD Investigators and sup- ported by the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK). The data from the MDRD reported here were supplied by the NIDDK Central Repositories. This manuscript was not prepared in collaboration with the investigators of the MDRD study and does not necessarily reflect the opinions or views of the MDRD study, the NIDDK Central Repositories, or the NIDDK.

Statement of Ethics:

The work in this manuscript was performed on de-identified participants of the MDRD study and was deemed to be IRB exempt.

Abstract

Background: Understanding factors which predict progression of renal failure is of great interest to clinicians.

Objectives: We examined machine learning methods to predict the composite outcome of death, dialysis or doubling of serum creatinine using the modification of diet in renal disease (MDRD) data set.

Methods: We specifically evaluated a generalized linear model, a support vector machine, a decision tree, a feed-forward neural network and a random forest evaluated within the context of 10 fold validation using the CARET package available within the open source architecture R program.

Results: We found that using clinical parameters available at entry into the study, these computer learning methods trained on 70% of the MDRD population had prediction accuracies ranging from 66-77% on the remaining 30%. Although the support vector machine methodology appeared to have the highest accuracy, all models studied worked relatively well.

Conclusions: These results illustrate the utility of employing machine learning methods within R to address the prediction of long term clinical outcomes using initial clinical measurements.

Keywords

hypertension, blood pressure, chronic renal disease, correlation, machine learning, cardiovascular disease

Introduction

The modification of diet in renal disease study was a landmark clinical trial examining the effectiveness of blood pressure control and dietary protein restriction on renal disease progression.[1] Although the maneuvers studied in the project were not very successful at attenuating renal disease progression, the most commonly used formula for estimating glomerular filtration rate (eGFR) was developed from this study. We chose to use this data set to examine whether we could predict outcomes using different mathematical methodologies on this population.

Methods

A retrospective study was performed using data acquired in the “Modification of Diet in Renal Diseases” or MDRD study.[2] Results from this study have been reported elsewhere.[1-5] This data containing 25,903 records was imported into R Studio and simplified into 840 unique patient records. Within this data, we found 692 subjects who had complete records for 76 variables determined on the initial visit which were used for modeling (Appendix 1). The outcome measurement used was a composite variable consisting of death, dialysis or a doubling of the serum creatinine.[6]

All analysis was performed using the open source program R. We used a generalized linear (logistic regression) model as our default.[7] In addition, we examined the utility of a support vector machine which involves the multi-dimensional sorting of data based on the development of a "hyperplane" which effectively separates the data.[8] We also examined the performance of decision trees with the RPART package and random forests with the randomForest package.[9,10] The decision tree approach utilized three or more decisions. With the random forest technique, we found that the optimal number of trees was around nine.   Different feed forward neural network architectures were explored using the nnet and neuralnet packages.[11] We found optimal performance with one hidden layer containing ten hidden neurons after this exploration. The CARET package was used for comparison of the mature models employing ten folds and three repeats.[12] Other packages within R were used for different specific tasks (e.g., rminer to determine relative importance of variables, nnet for construction of the neural network, randomForest (RFor) for constructing random forests).[11,13-17] Representative R routines for "cleaning the data" (e.g., centering and scaling, Appendix 2) splitting the data into training (70%) and testing (30%) sets, and comparing the different models with the categorical output (Appendix 3) are attached.

Results and Discussion

The MDRD study is famous for yielding clinical estimates of glomerular filtration rate, but it should be emphasized that it was developed to test whether dietary protein restriction would ameliorate the progression of renal failure. This study has been reviewed extensively elsewhere, but for the purpose of our interest, we had a group of patients with some degree of chronic kidney disease who developed a composite endpoint consisting of death, dialysis or a doubling of the baseline creatinine. Ergo, it was possible to fit the baseline data with different models.

We found that each of the models studied had some success at prediction. It turns out that for each of the models, specificity was superior to sensitivity and accuracy ranged between 66 and 77%. When we examined the Receiver Operator Curves (ROC, Figure 1), it appears that the SVM and the RFor models performed somewhat better than the other models. When we examined which variables were most important in these models with the rminer package (Figure 2), we found that the baseline serum creatinine was featured in the top three variables (ranked in descending order) in all of the models and was the top variable in the GLM, the SVM and the RFor models. This is not terribly surprising as the initial renal function would be expected to predict outcomes in this population with chronic kidney disease. Of interest, dietary protein and blood pressure did not achieve great importance in these different models. Again, this was not surprising as these interventions did not significantly affect outcomes. Another point to emphasize was that each of the models we used did relatively well (Table 1). While we emphasized that all of the analysis was done within the context of ten fold validation with averaging on the training set within the CARET package (see Appendix 3), the truth is that this didn't seem to make much difference for any of the aforementioned models which performed almost identically when just trained on the training set without ten fold validation. Variations in the relative size of the training and testing sets (varying from 50:50 to 85:15) also did not significantly affect our results (data not shown).

Figure 1: Receiver operator curves (ROC) showing sensitivity against 1-specificity for generalized linear model (GLM) - red color, area under curve (AUC) = 0.59, support vector machine (SVM) – green color, AUC=0.77, decision tree (RPART) – blue color, AUC=0.64, neural network (NNet) – orange color, AUC= 0.67, random forest (RFor) – purple color, AUC= 0.72 developed with the training set (70% of total) and applied to the testing set (30% of total) using a categorical output.

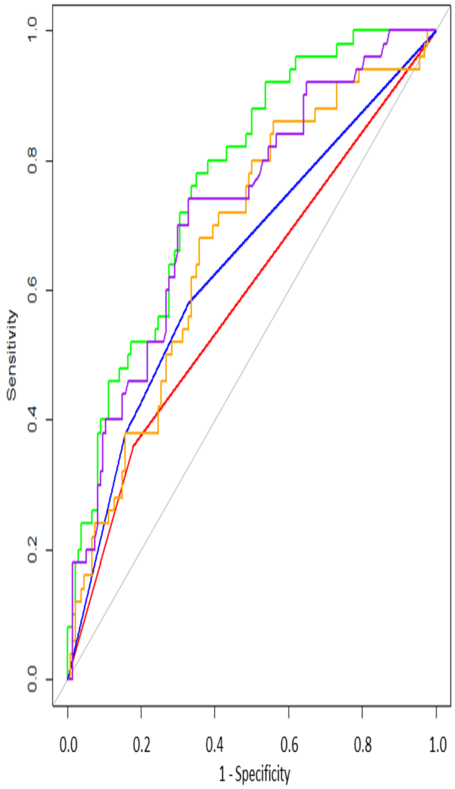
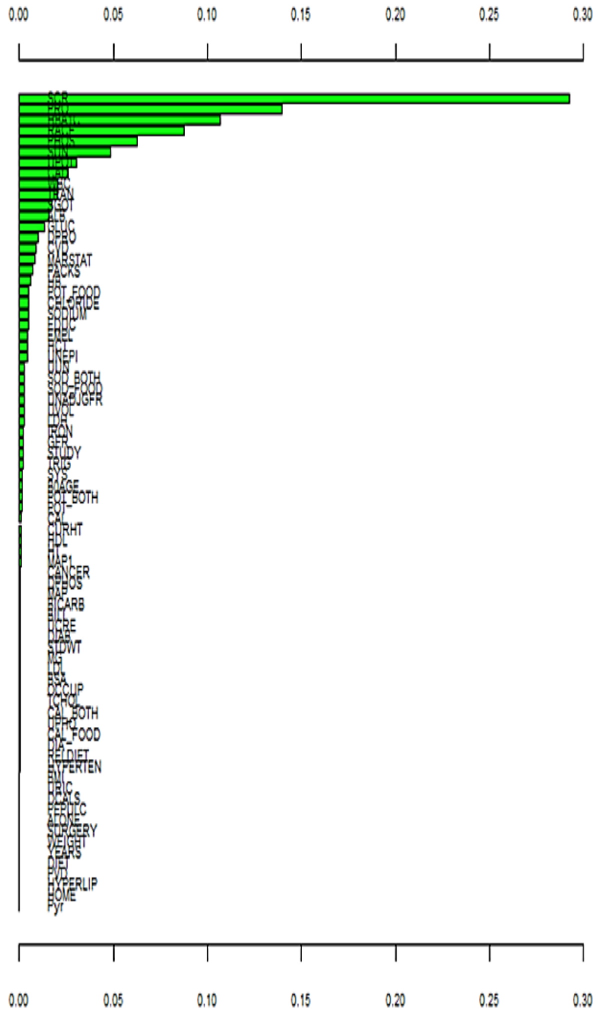
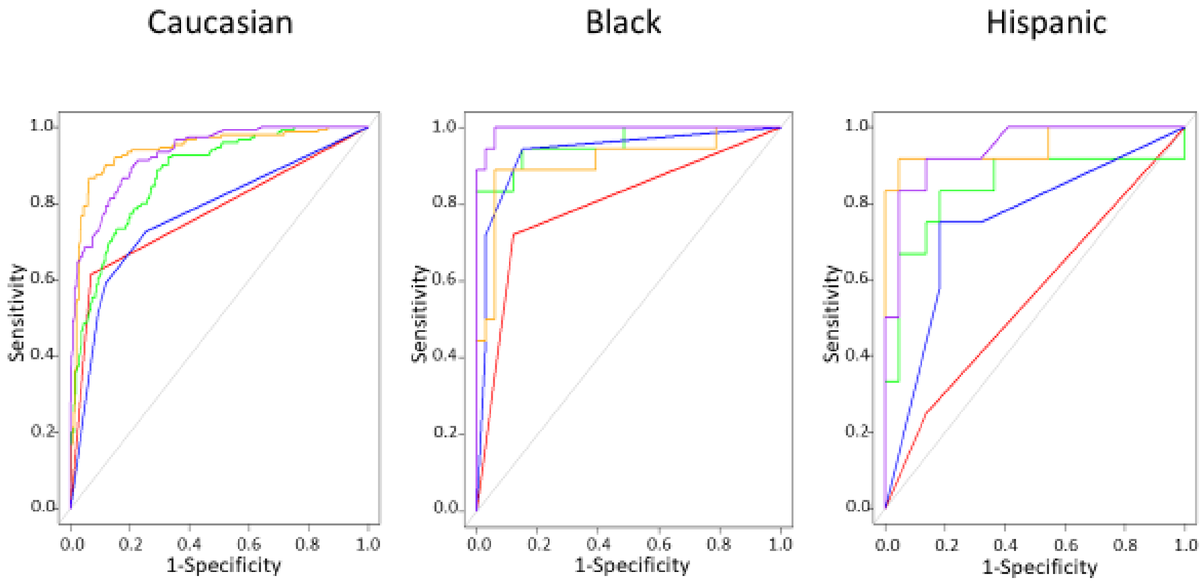


Figure 2: Relative importance of variables in the SVM model. Similar plots were produced and analyzed for all of the models studied. For all but the neural network model, the top three variables accounted for the vast majority of the model. The top three variables in importance for each model were as follows. GLM: SCr (serum [creatinine]), GFR (glomerular filtration rate) and Pro (proteinuria); SVM: SCr, Pro and Bicarb (serum [bicarbonate]); RPART: SCr, pack-years, and Pro; NNet: UPot (urinary [potassium], Packs (packs of cigarettes/day) and SCr, and RFor: SCr, GFR and Prot.



Among subjects that had complete records, 591 were Caucasian, 51 were Black and 34 identified themselves as Hispanic (the remaining 16 were spread among other categories). To examine whether the models developed on the training set described above performed well with different racial groups, we looked at the performance on the Caucasian, Black and Hispanic subsets. We found that the predictive models performed similarly across the different racial subsets (Figure 3). It is important here to point out that the predictive value of these models was superior in these racial subsets to that achieved in the aforementioned randomly selected testing set, in part, because they were tested on some patients who were in the original training set. Therefore and due to these difficulties, ethnicity is an area that shows promise but will be explored later in another study with different dataset.

Figure 3: Receiver operator curves (ROC) showing sensitivity against 1-specificity for generalized linear model (GLM) - red color, support vector machine (SVM) – green color, decision tree (RPART) – blue color, neural network (NNet) – orange color, random forest (RFor) – purple color, developed with the training set (70% of total) and applied to testing set consisting of all Caucasian, Black and Hispanic patients, respectively. Note that some patients in these racial groups were used in both the training and testing sets. Although the linear model did not perform well in the Hispanic subset, other models especially the random forest, neural network and support vector machine models performed extremely well in all racial subsets.



These data are of interest for several reasons. First, they show that creation of several different prediction models with clinical data sets is relatively straightforward within the open source architecture of R. Second, these data demonstrate that all of these models perform relatively well and end up "choosing" the same key clinical elements to predict clinical outcomes. Moreover, the models validate the clinical impression that knowing the severity of patient's renal failure is an excellent predictor of a composite clinical outcome which is weighted toward renal functional deterioration. For future projects, we recommend expansion of these models to include other clinical variables not included in the MDRD study which are known to reflect CKD progression.

References

1. Levey AS, Greene T, Schluchter MD, Cleary PA, Teschan PE, Lorenz RA, et al. Glomerular filtration rate measurements in clinical trials. Modification of Diet in Renal Disease Study Group and the Diabetes Control and Complications Trial Research Group. Journal of the American Society of Nephrology : JASN. 1993;4(5):1159-71.
2. Levey AS, Gassman JJ, Hall PM, Walker WG. Assessing the progression of renal disease in clinical studies: effects of duration of follow-up and regression to the mean. Modification of Diet in Renal Disease (MDRD) Study Group. Journal of the American Society of Nephrology : JASN. 1991;1(9):1087-94.
3. Levey AS, Greene T, Sarnak MJ, Wang X, Beck GJ, Kusek JW, et al. Effect of dietary protein restriction on the progression of kidney disease: long-term follow-up of the Modification of Diet in Renal Disease (MDRD) Study. Am J Kidney Dis. 2006;48(6):879-88.
4. Levey AS, Adler S, Caggiula AW, England BK, Greene T, Hunsicker LG, et al. Effects of dietary protein restriction on the progression of advanced renal disease in the Modification of Diet in Renal Disease Study. Am J Kidney Dis. 1996;27(5):652-63.
5. Levey AS, Berg RL, Gassman JJ, Hall PM, Walker WG. Creatinine filtration, secretion and excretion during progressive renal disease. Modification of Diet in Renal Disease (MDRD) Study Group. Kidney international Supplement. 1989;27:S73-80.
6. Levey AS, Greene T, Beck GJ, Caggiula AW, Kusek JW, Hunsicker LG, et al. Dietary protein restriction and the progression of chronic renal disease: what have all of the results of the MDRD study shown? Modification of Diet in Renal Disease Study group. Journal of the American Society of Nephrology: JASN. 1999;10(11):2426-39.
7. CA G, MJ M, JI S, BL M. Predicting Medical Student Success on Licensure Exams. Med Sci Educ. 2015;25:447-53.
8. Tirelli T, Gamba M, Pessani D. Support vector machines to model presence/absence of Alburnus alburnus alborella (Teleostea, Cyprinidae) in North-Western Italy: comparison with other machine learning techniques. C R Biol. 2012;335(10-11):680-6.
9. Chen T, Cao Y, Zhang Y, Liu J, Bao Y, Wang C, et al. Random forest in clinical metabolomics for phenotypic discrimination and biomarker selection. Evid Based Complement Alternat Med. 2013;2013:298183.
10. Khondoker MR, Bachmann TT, Mewissen M, Dickinson P, Dobrzelecki B, Campbell CJ, et al. Multi-factorial analysis of class prediction error: estimating optimal number of biomarkers for various classification rules. J Bioinform Comput Biol. 2010;8(6):945-65.
11. Zhang Z. A gentle introduction to artificial neural networks. Ann Transl Med. 2016;4(19):370.
12. Tsiliki G, Munteanu CR, Seoane JA, Fernandez-Lozano C, Sarimveis H, Willighagen EL. RRegrs: an R package for computer-aided model selection with multiple regression models. J Cheminform. 2015;7:46.
13. Liu R, Li X, Zhang W, Zhou HH. Comparison of Nine Statistical Model Based Warfarin Pharmacogenetic Dosing Algorithms Using the Racially Diverse International Warfarin Pharmacogenetic Consortium Cohort Database. PLoS One. 2015;10(8):e0135784.
14. Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez JC, et al. pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics. 2011;12:77.
15. Emir B, Johnson K, Kuhn M, Parsons B. Predictive Modeling of Response to Pregabalin for the Treatment of Neuropathic Pain Using 6-Week Observational Data: A Spectrum of Modern Analytics Applications. Clin Ther. 2017;39(1):98-106.
16. Hengl T, Mendes de Jesus J, Heuvelink GB, Ruiperez Gonzalez M, Kilibarda M, Blagotic A, et al. SoilGrids250m: Global gridded soil information based on machine learning. PLoS One. 2017;12(2):e0169748.
17. Gallo S, Hazell T, Vanstone CA, Agellon S, Jones G, L'Abbe M, et al. Vitamin D supplementation in breastfed infants from Montreal, Canada: 25-hydroxyvitamin D and bone health effects from a follow-up study at 3 years of age. Osteoporos Int. 2016.

Table 1: Confusion Matrices with Different Models

Model	**Yes**	**No**	**Specificity**	**Sensitivity**	**Accuracy**
Reference	**50**	**134**			
GLM	18/50	100/134	82%	36%	64%
SVM	23/50	119/134	89%	46%	77%
RPart	21/50	108/134	81%	42%	70%
NNet	19/50	102/134	76%	38%	66%
RForest	20/50	114/134	85%	40%	73%

Sensitivity refers to true positives divided by the sum of true positives and false negatives.
Specificity refers to the true negatives divided by the sum of true negatives and false positives.
Accuracy is calculated on the testing set as the fraction of all assignments which are correct.

Appendix 1:

Var Name	Var Description	Var Name	Var Description
"STDWT"	Ideal Weight	"UNADJGFR"	GFR not adjusted
"CURHT"	Height	"BSA"	Body Surface Area
"WEIGHT"	Weight	"HT"	Height during study
"BMI"	Body Mass Index	"RACE"	Race
"GFR"	Glomerular Filtration Rate	"EDUC"	Education level
"MAP1"	Mean Arterial Pressure 1	"OCCUP"	Occupation code
"UCRE"	Urinary [Creatinine]	"HOME"	Stay at Home
"UUN"	Urinary [Urea Nitrogen]	"EMPL"	Employment
"UPHO"	Urinary [Phosphate]	"RELDIET"	Diet Group
"UVOL"	Urine Volume	"MARSTAT"	Marital Status
"UPOT"	Urine [Protein]	"ALONE"	Live Alone
"SUN"	Serum Urea Nitrogen	"DIAB"	Diabetic
"SCR"	Serum Creatinine	"CAD"	Coronary Artery Disease
"TCHOL"	Total Cholesterol	"PEPULC"	Peptic Ulcer
"TRAN"	Transferrin	"CANCER"	Cancer
"ALB"	Albumin	"CVD"	Stroke
"HBA1C"	HBA1C	"PVD"	Peripheral Vascular Disease
"PHOS"	Serum Phosphate		Hypertension
"TRIG"	Serum Triglycerides	"HYPERTEN"	Hyperlipidemia
"LDL"	Low Density Lipoprotein	"HYPERLIP"	Prior Surgery
"HDL"	High Density Lipoprotein	"SURGERY"	Smoking packs per day
"POT"	Serum Potassium	"PACKS"	Years smoking
"BICARB"	Serum Bicarbonate	"YEARS"	Product of prior two
"CAL"	Serum Calcium	"Pyr"	Serum Sodium
"MG"	Serum Magnesium	"SODIUM"	Serum Chloride
"HB"	Hemoglobin	"CHLORIDE"	Serum Uric Acid
"HCT"	Hematocrit	"URIC"	Serum Bili
"DPRO"	Dietary Protein	"BILI"	Serum LDH
"DCALS"	Dietary Calcium	"LDH"	Alanine Transaminase
"DPHOS"	Dietary Phosphate	"SGOT"	Serum Glucose
"IRON"	Serum Iron	"GLUC"	Potassium from food
"WBC"	White Blood Cells	"POT_FOOD"	Total Potassium
"MAP"	Mean Arterial Pressure	"POT_BOTH"	Sodium from food
	during Study	"SOD_FOOD"	Total Sodium
"UNEPI"	UNEPI	"SOD_BOTH"	Calcium from food
"STUDY"	Study Group	"CAL_FOOD"	Total Calcium
"DIET"	Diet Study	"CAL_BOTH"	Age at randomization
"PRO"	Protein Study	"B0AGE"	
"SYS"	Systolic Blood Pressure		
"DIA"	Diastolic Blood Pressure		

Appendix 2: Cleaning Data

```
#call in data set, remove patient index variable
xx=x[2:86]
# only complete cases
xx=xx[complete.cases(xx),]
dim(xx)

#create yes no variable for outcome
k=NULL
for(i in 1:dim(xx)[1]){
  if(xx$EV_ALL[i]>0){
    k[i]="yes"
  }else{
    k[i]="no"
  }
}
#create set for analysis
z=xx[,2:77]
z=cbind(z,k)
colnames(z)[77]="output1"
#scale and center data
zz=preProcess(z,c("center","scale"))
z=predict(zz,z)
```

Appendix 3: ROC curve and model analysis

```
#load libraries
library(ROCR)
library(pROC)
library(rpart)
library(caret)
library(nnet)
library(C50)
library(ggplot2)
library(lattice)
library(randomForest)
library(rminer)

# produce copy in a text file
sink('output1_2.txt', split=TRUE)

# separate the “cleaned” dataset z randomly into training and testing sets
set.seed(2)
ind = sample(2, nrow(z), replace = TRUE, prob = c(0.75, 0.25))
trainset = z[ind == 1,]
testset = z[ind == 2,]

# train the different models within CARET on the training set
control = trainControl(method = "repeatedcv", number = 10, repeats = 3, classProbs = TRUE,
summaryFunction = twoClassSummary)

glm.model = train(output1 ~ ., data = trainset, method = "glm", metric = "ROC", trControl =
control)

svm.model = train(output1 ~ ., data = trainset, method = "svmRadial",metric = "ROC", trControl
= control)

rpart.model = train(output1 ~ ., data = trainset, method = "rpart", metric = "ROC", trControl =
control)

tunGrid=expand.grid(size=c(5),decay=c(0.1))
nnet.model = train(output1 ~ ., data=trainset, method = "nnet", metric="ROC",
trControl=control, tuneGrid=tunGrid)

rfor.model = train(output1 ~ ., data=trainset, method = "rf", metric="ROC", trControl=control)

# establish predictions from these models on the testing set
glm.probs = predict(glm.model, testset[,! names(testset) %in% c("output1")], type = "prob")
svm.probs = predict(svm.model, testset[,! names(testset) %in% c("output1")], type = "prob")
```

```
rpart.probs = predict(rpart.model, testset[,! names(testset) %in% c("output1")], type = "prob")
nnet.probs=predict(nnet.model,  testset[,! names(testset) %in% c("output1")], type = "prob")
rfor.probs=predict(rfor.model,  testset[,! names(testset) %in% c("output1")], type = "prob")

#create receiver operator curves
windows()

glm.ROC = roc(response = testset[, c("output1")], predictor = glm.probs$yes, levels =
levels(testset[, c("output1")]))
plot(glm.ROC,add=F, col =" red")

svm.ROC = roc(response = testset[, c("output1")], predictor = svm.probs$yes, levels =
levels(testset[, c("output1")]))
plot(svm.ROC, add = TRUE, col ="green")

rpart.ROC = roc(response = testset[, c("output1")], predictor = rpart.probs$yes, levels =
levels(testset[, c("output1")]))
plot(rpart.ROC, add = TRUE, col ="blue")

nnet.ROC=roc(response = testset[, c("output1")], predictor = nnet.probs$yes, levels =
levels(testset[, c("output1")]))
plot(nnet.ROC, add = TRUE, col ="orange")

rfor.ROC=roc(response = testset[, c("output1")], predictor = rfor.probs$yes, levels =
levels(testset[, c("output1")]))
plot(rfor.ROC, add = TRUE, col ="purple")

#produce confusion matrices

glm.pred=predict(glm.model,testset[,!names(testset)%in% c("output1")])
table(glm.pred,testset[,c("output1")])
confusionMatrix(glm.pred,testset[,c("output1")])

svm.pred=predict(svm.model,testset[,!names(testset)%in% c("output1")])
table(svm.pred,testset[,c("output1")])
confusionMatrix(svm.pred,testset[,c("output1")])

rpart.pred=predict(rpart.model,testset[,!names(testset)%in% c("output1")])
table(rpart.pred,testset[,c("output1")])
confusionMatrix(rpart.pred,testset[,c("output1")])

nnet.pred=predict(nnet.model,testset[,!names(testset)%in% c("output1")])
table(nnet.pred,testset[,c("output1")])
confusionMatrix(nnet.pred,testset[,c("output1")])

rfor.pred=predict(rfor.model,testset[,!names(testset)%in% c("output1")])
```

```
table(rfor.pred,testset[,c("output1")])
confusionMatrix(rfor.pred,testset[,c("output1")])

#determine variable importance in different models

model_rpart=fit(output1~., trainset, model="dt")
VI_rpart=Importance(model_rpart,trainset,method="sensv")
L_rpart=list(runs=1,sen=t(VI_rpart$imp), sresponses=VI_rpart$sresponses)
windows()
mgraph(L_rpart,graph="IMP",leg=names(trainset),cex=0.6,col="blue")

model_rfor=fit(output1~., trainset, model="randomForest")
VI_rfor=Importance(model_rfor,trainset,method="sensv")
L_rfor=list(runs=1,sen=t(VI_rfor$imp), sresponses=VI_rfor$sresponses)
windows()
mgraph(L_rfor,graph="IMP",leg=names(trainset),cex=0.6, col="purple")

model_glm=fit(output1~., trainset, model="cv.glmnet")
VI_glm=Importance(model_glm,trainset,method="sensv")
L_glm=list(runs=1,sen=t(VI_glm$imp), sresponses=VI_glm$sresponses)
windows()
mgraph(L_rfor,graph="IMP",leg=names(trainset),cex=0.6,col="red")

model_nn=fit(output1~., trainset, model="mlpe")
VI_nn=Importance(model_nn,trainset,method="sensv")
L_nn=list(runs=1,sen=t(VI_nn$imp), sresponses=VI_nn$sresponses)
windows()
mgraph(L_nn,graph="IMP",leg=names(trainset),cex=0.6,col="orange")

model_svm=fit(output1~., trainset, model="svm")
VI_svm=Importance(model_svm,trainset,method="sensv")
L_svm=list(runs=1,sen=t(VI_svm$imp), sresponses=VI_svm$sresponses)
windows()
mgraph(L_svm,graph="IMP",leg=names(trainset),cex=0.6,col="green")
```